"□□□——□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"
□□□AI□□□□
□□
"Artificial Intelligence Neural Network Adaptive Self-induction Self-feedback Stream of Consciousness Absorption Integration Purification Sublimation" 2025v1.1 Global Multilingual Online Edition E-book artificial intelligence technology research and development innovation peak.

« Réseau neuronal d'intelligence artificielle adaptative à l'induction de l'auto-réponse du flux de conscience absorbe l'intégration, l'intégration, la purification et la sublimation » 2025v1.1 Version Web multilingue mondiale eBook R&D de la technologie de l'intelligence artificielle au sommet de l'innovation

«Адаптивные нейронные сети искусственного интеллекта «Адаптивные индукции, самообменные потоки сознания» 2025v1.1 Глобальная многоязычная сетевая версия электронной книги «Искусственный интеллект научно-исследовательская и инновационная технология»

"Red neuronal de Inteligencia Artificial Adaptive Induction Self Feedback Flujo de conciencia de absorción e integración de integración, purificación y sublimación" 2025v1.1 Edición web multilingüe global eBook investigación e innovación de la tecnología de inteligencia artificial

●●□□ □□□□□□□□□□□□□□□ 1. □□□□□□□□- □□□□□□CNN□□□- □□□□□□RNN/LSTM/GRU□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NLP□□□□□□□- Transformer □□□□□□□□□□□□□□□□□□□□□□□□□□NLP□□GPT□□□□□□□□□□□BERT□CLIP□□□□□□□- □□□□□GNN□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2. □□□□□□□- □□□□□Backpropagation□□□□□□□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□□□□□□□□□□□□□□□MSE□□□□□□□Adam□SGD□□□□□□RMSprop□□- □□□□□□Dropout□L1/L2□□□□Batch Normalization□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ 1. □□□□□□□- □□□□□□□□□□□□□□□□□□□CLIP□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□□Cross-Attention□□□□□□□□□Multi-modal Transformer□□□□□□□□□□- □□□□□□□GPT-4V□□□□GPT□□FLAVA□MDETR□□□□□□□□□□□□□□□□□□□□□2. □□□□□- □□□□□□CV□□□□□□□YOLO□Faster R-CNN□□□□□□□Mask R-CNN□□3D□□□□□□□□□□□□□□□□- □□□□□□□□□□□□ASR□□Whisper□□□□□□□TTS□□Tacotron□□□□□□□- □□□□□□NLP□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ 1. □□□/□□□□□- □□□□□Contrastive Learning□□□□□□□□□□□□□□SimCLR□MoCo□□□□□□□□□□□□□□□□□- □□□□□□GAN□□□□□□□□□□□□□□□□□□□□StyleGAN□Diffusion Models□□□□□□□- □□□□□□□□□□□□□□□□□□BERT□□□□□□□□AE□□□□□□□□□□2. □□□□RL□□□□□□□- □□□□□□DRL□□□□CNN/Transformer□DRL□□□□DQN□PPO□SAC□□□□□□□□□□□□□□□□□□- □□□- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□NAS□□ □□□□□□□□□□□ 1. □□□□□□SNN□- □□TrueNorth□□2. □□□□BCI□□□□□□- □□□□BCI□□□EEG□fMRI□□□□□□□□□□□□□□□□□□□□□□□□- □□□BCI□□□□□□□□□□□□□□□□□Neuralink□□□□□□□□□□□□□□□3. □□□□□- □□□□□□□□□□□□□□□□□□□□□□□□□HTM□□□□□□□□Neocognitron□□ □□□□□□□□□□□□□ 1. □□□□□□□XAI□-

□□□□□□□□□□□□Transformer□□□□□□□□□□□BERT□□□□□□□□- □□□□□□
SHAP□SHapley Additive exPlanations□□LIME□Local Interpretable Model-agnostic Explanations□□□□□□□□□□□□- □□-□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2. □□□□- □□□□□□Pearl□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□1. □□□□□□-GPU/TPU/NPU□NVIDIA GPU□CUDA□□□□Google TPU□□□□□□□□□□□□□AI□□□□□□- □□□□□□□□“□□□”□□□□Graphcore IPU□□□□□□□□□□□□- □□□□□□□□□□□□□□□□Intel Loihi□IBM TrueNorth□2. □□□□□□□□□□- □□□□□□□Quantization□□□□□Pruning□□□□□□Knowledge Distillation□□□□□□□□□□□□□□□□□□- □□□□□□TensorRT□ONNX Runtime□MNN□□□□□□□□□□□□□□□ □□□□□□□□□□1. □□□□□- □□□□□□□□□□□□□□□□□□LiDAR□□IMU□□□□□□□□□□□□□□□□BEV/Transformer□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□I3D□LSTM-CNN□□□□□2. □□□□□□- □□□□□□□□□AutoAugment□□□□□□□□FGSM□□□□□□□□□ □□□□□□□□□□□□1. □□□□□□- □□□□□□MPC□□DRL□□□□□□□□□□□□□□□□□□□□□- □□□PID□□□□□□□□□□□□□PID□□□□□□□□□□□2. □□□□□- □□□□□□□□□Model-Based RL□□ □□□□□□□□1. □□□□□- □□□□Adversarial Training□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□CleverHans□□□□□□□□□2. □□□□□- □□□□Federated Learning□□□□□□□□□□□□□□□□□□□□□□□□- □□□□Differential Privacy□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□1. □□□□□□□□- □□□□□□□□□□GWT□□□□□□□□□□□□□□□□□□□□□□□□- □□2. □□□□□□□□□- □□□□Meta-Learning□□□□□□“□□□□”□□□□□□□□□□□□□□□□□MAML□□□□- □□□□□□□□□□□□□□□□□Curiosity-Driven Learning□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ □□ - □□□□□□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□□□□□□□□□□GPT□□□□□□□□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

●□□□①□□□②□□A. □□□□□□B.□□□□□□C.□□□□□□□□□□□D.□□□□□E.□□□□□□F.□□□□□D.□□H.□□□M.□□

●□□---### □□□□□□□□□□□□□□□□1. **□□□□□□□□□**- **□□□□□□□**□□□□□□□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□GPT-4o□□□□□□□□□□□□□□□□□□□□□□□□□□ - **□□□□□□□**□□-□□□□□□□□□□□□□□□□□□□2. **□□□□□□□□**- **□□□□□□□□□**□□□□□□□BCI□□□□□□□□□□□□□□-□□□□□□□□□□Neuralink□□ - **□□□□□**□□□□□□□□□□□□□□□□□MPC□□□□□□□□□DRL□□□□□□□□□+□□□□□

[illegible]---### [illegible] 1. **[illegible]** - **[illegible]** [illegible] CLIP [illegible] - **[illegible]** [illegible] Neuro-Symbolic AI [illegible] 2. **[illegible]** - **[illegible]-[illegible]** [illegible] IBM TrueNorth [illegible]-[illegible] TinyML [illegible] - **[illegible]** [illegible] VaR [illegible]---### [illegible] 1. **[illegible]** - **[illegible]** [illegible] Domain Adaptation [illegible] X [illegible] - **[illegible]** [illegible] Grad-CAM [illegible] Do-Calculus [illegible] 2. **[illegible]** - **[illegible]** [illegible] Intel Loihi [illegible] - **[illegible]** [illegible]---### [illegible] 1. **[illegible]** - [illegible] 2. **[illegible]** - [illegible] AI [illegible]---### [illegible]-[illegible]-[illegible]·[illegible]

●[illegible] AI [illegible] **[illegible]** [illegible]---### **[illegible]"[illegible]-[illegible]-[illegible]-[illegible]"[illegible]** [illegible] AI [illegible] 1. **[illegible]** [illegible] - **[illegible] AI** [illegible]/[illegible]"[illegible]"[illegible] - **[illegible]** [illegible] **[illegible]** [illegible] **[illegible]** [illegible] **[illegible]+[illegible]+[illegible]** [illegible] 2. **[illegible]** [illegible] A-G [illegible] ```mermaid graph LR A[[illegible]] --> B([illegible]) B --> C{[illegible]} C --> D[[illegible]-[illegible]] C --> E[[illegible]-[illegible]] C --> F[[illegible]-[illegible]] F --> G[[illegible]/[illegible]] G --> H[[illegible]] ``` - **[illegible]** [illegible] **"[illegible]"** [illegible]+[illegible]---### **[illegible]** [illegible] 1. **[illegible]** [illegible] - [illegible] <70% [illegible] - [illegible] >200ms [illegible] 2. **[illegible]** [illegible] - **[illegible]** [illegible] **[illegible]** [illegible]/[illegible]/[illegible] - **[illegible]** [illegible] **[illegible]**---### **[illegible]** [illegible] **"[illegible]"** [illegible] AI [illegible]| **[illegible]** | **[illegible]** | **[illegible]** ||----------------|---------------------------------|--------------------------|| **[illegible]** | [illegible]+[illegible] | [illegible] || **[illegible]** | [illegible]+[illegible] | [illegible] || **[illegible]

●[illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]

□□

● □□---### □□□□□□□□□□□□□□□□□□ 1. **□□□□□□□□□** - **□□□□□□□**□□□□□□□□□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□GPT-4o□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ - **□□□□□□□**□□-□□□□□□□□□□□□□□□□□□□2. **□□□□□□□□□** - **□□□□□□□□□**□□□□□□□□BCI□□□□□□□□□□□□□-□□□□□□□□□□□Neuralink □□ - **□□□□□□**□□□□□□□□□□□□□□□□□□MPC□□□□□□□□□DRL□□□□□□□□□+□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□---### □□□□□□□□□□□□□□□□ 1. **□□□□□□□□□** - **□□□□□□□□**□□□□□□□□□□□□□□□□□□□□□□□□□□□CLIP □□□ - **□□□□□□□□□**□□□Neuro-Symbolic AI□□□□□□□□□□□□□□2. **□□□□□□□□□□** - **□□-□□□□□□**□□□□□□□□□□□□ IBM TrueNorth□□□□□-□□□□□□□□□□□□□□□□□ TinyML□□□□□□□□ - **□□□□□□□□**□□□□□□□□□□□□□□□□□□□□□ VaR □□---### □□□□□□□□□□□□□ 1. **□□□□□□□□□□□** - **□□□□□□□□**□□□□□□□□□□□□□□□□□□□Domain Adaptation□□□□□ X □□□□□□□□□□□□□□□□□□□□□□□□□□□□ - **□□□□□**□□□□□□□□□□□□ Grad-CAM□□□□□□□□□□□ Do-Calculus□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2. **□□□□□□□□□** - **□□□□□□**□□□□□□□□□ Intel Loihi□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ - **□□□□□□□**□□□---### □□□□□□□□□□□□ 1. **□□□□□□□□** - □□□2. **□□□□□□□□** - □□ AI □□□□□---### □□-□□-□□□·□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

●□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□“□□□□□□□□□□□□□□□”□□□ 1. □□□□□□□□□□- □□□□□□□□□□□□□□ Neuralink □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□30dB□□□□□□□□□□□□□□□□□□□□□□□□□□ 10%-15%□□□□□□□□□□□□□□□□□□□□□□□□PEDOT:PSS □□□□□□□□□□□□□□□□□□□□ Transformer □□□□□□□□□- □□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□ 10 □□-□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 10^4 □□□□□□□□2. □□-□□□□□□□□□□- “□□□□□□”□□□ DARPA □ CRAFT □□ GPT-4□□ 42%□□□□□□□□□□□□□□□□□□□ 200ms□□□□□□□□- □□“□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□Sparse Autoencoder□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 0.01g □□□□□□□□□□□□□□□□□□□□□□□□□□-□□□ 20ms □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□ “A-F □□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1. □□□□□□□□□□- □□□-□□□□□□□□□□MIT □ Media Lab □□□□□□□□□□□Event Camera□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 0.5 □□□□□□□□□□□□□□□□□□□□2000fps vs □□ 30fps□□- □□“□□□□□□□”□OpenAI □ Sora □□□□□□□□□□□□□□□□□□□□□□□□CLIP □□ 0.72□□□□□□□□□□□□□□ 3 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Probabilistic Scene Graphs□□2. □□□□□□□□□□- □□DeepMind □ Gato □□□□□ 600+□□ 15%□□□□□□□□ MERLION □□□□□□□□□□□Dynamic Routing□□□□□□□□□□□□ 8%□□□□□□□□□30%□- “□□□□□□□□□□□□□□□”□□□□□□□□□□□□Waymo □ HydraNet □□□□□□□□□□□□□□□□□□

□mAP 92.3%□□□□□□□□□<0.01 □/□□ NuSMV □□□□□□3. □□□□□□□□□□- □□□□□□□□□□□□□□□□□□ Transformer□□□□□□□□□Counterfactual Reasoning□□□□□□□□□□□□□□□□□ Judea Pearl □□□□□□□□□□□Causal Bayesian Networks□□□□□□□□□□□□□□□□□□□□□ATE □□□□<5%□□□□□□□□□□□□□□□□□□□□□>1000 □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□"□□□□□□□"□□□ 1. □□□□□□□□□- □□□ Optimus □□□□□□ 40 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1mph□□□□□□□□□□□□□ 0.1N□□□□□□□□□□□□□□□□□□<10ms□□□□□□□□□□□□□45 □□□□□□□□□□□- □□□□□□□□□□□□□□□ Digital Twin □□□□□□□□□□□□□□□□□□<50ms□□□□□□□□□□□□37%□□□□□□□□□□□□□□□□□□□□ 10^4 □□□□□□□□ 5G+□□□□□□□□□□□□<10ms□□□□>99.999%□□2. □□□□□□□□□□- □□□□□□□DRL□□□□□□□□□□□□□□ 30%□□□□□□□□□□□□□□□□□□□□□□-□□-□□□□□□□□□□□□□ DRL □□□□□□□ 10^6 □□□□□□□□□□□□□□□□□□□□□□ AutoRL □□□□□□□□Meta-Learning□□□□□□□□ 10 □□□□□□□□□□□□□□□□□□□□□□□□□□- "□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□ Waymo □ ChauffeurNet□□□□□□□□□□□□□□□□□□□□ 0.01 □/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□+□□□□□□□□□□□□□□□□□□□□□□□□□□□□<200ms□□□□□□□3. □□□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 9 □□/□□□□□□□□□□□□□□□□□SSVEP□□□□□□□□□□□□□□□□□□□□□□□ 95%□□- □□□□□□□□□□□□□T20 □□□□□□□□□□□□□□□□□□□□□□□ 92%□□□□□□□□□□□□□□□□ 1200+□□ □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□"□□□□□□□□□□□"□□□□□□□□□□□□□□□□ 1. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Global Workspace Theory□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□2. □□□□□□□□□□·□□□□□□□□□□□Memory Wall□□□□□□□□□□□□□□□ 90%□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ IBM □ TrueNorth □□□□□□□ 10^4 TOPS/W□□3. □□□"□□□□□□□□□□"□□□□ AI □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□ - □□□□□□□□□□□□□□□ fMRI □□- □□□□□□□□□□□□□□□□□□□□<1ns□□- □□□□□□□□□□ Human Brain Project □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Allen □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□——□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□"□□"□□□□□□□□□□"□□□□□□□□□·□□□□□David Chalmers□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□——□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□● □□□□□□□□□□□□□ ○ □□□ ○ □□□● □□□□□ ○ □□□● □□□□□□□□ ○ □□□● □□□□□□□□ ○ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□CNN□□□□□□□□□□□□□□□□□□□LSTM□□□□□□□□□□□GRU□□□○ □□□● □□□□□□□□□ ○ □□● □□□□□□□□ ○ □□□□□□□□□□AutoML□□□NAS□□□● □□□□□ ○ □□

● ○

● ○

EWC SI

● ○

● ○

● ○

● ○

● ○

● AlexNet VGGNet ImageNet

AlexNet VGGNet

● Google DeepMind AlphaGo AlphaGo

● DreamGen AI

22 21%

45.5% 0 ●

ERA - 42

100 2

●

● IBM Watson

IBM Watson

●

● ChatGPT ○ ChatGPT

○

● ○

○

● IBM Watson for Oncology ○

○

● ○

X CT MRI

○

● ○

[illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible] ● [illegible] Autopilot ○ [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible] ● [illegible] MindSphere ○ [illegible] MindSphere [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible]

●●[illegible] ● [illegible] ● [illegible] ● [illegible] “[illegible]” [illegible] ● [illegible] ● [illegible] ● [illegible] ● [illegible] ● [illegible]

●[illegible] “[illegible]” [illegible] Python [illegible] import numpy as np# [illegible] class Sensor:

def __init__(self, name): self.name = name def get_data(self): # [illegible] return np.random.rand()# [illegible] class MultiSensorFusion: def __init__(self): self.sensors = [] def add_sensor(self, sensor): self.sensors.append(sensor) def fuse_data(self): data = [sensor.get_data() for sensor in self.sensors] # [illegible] fused_data = np.mean(data) return fused_data# [illegible] class AdaptiveFeedbackControl: def __init__(self, target_value): self.target_value = target_value self.control_factor = 0.1 def adjust(self, current_value): error = self.target_value - current_value adjustment = self.control_factor * error return adjustment# [illegible] class ConsciousnessPerception: def __init__(self): self.memory = [] def perceive(self, data): # [illegible] self.memory.append(data) def process(self): # [illegible] if self.memory: return np.mean(self.memory) return 0 def feedback(self, result): # [illegible] if result > 0.5: return 1 else: return 0# [illegible] if __name__ == "__main__": # [illegible] sensor1 = Sensor("Sensor1") sensor2 = Sensor("Sensor2") # [illegible] fusion = MultiSensorFusion() fusion.add_sensor(sensor1) fusion.add_sensor(sensor2) fused_data = fusion.fuse_data() # [illegible] target = 0.7 control = AdaptiveFeedbackControl(target) adjustment = control.adjust(fused_data) # [illegible] perception = ConsciousnessPerception() perception.perceive(fused_data) processed_result = perception.process() feedback_result = perception.feedback(processed_result) print(f"Fused data: {fused_data}") print(f"Adjustment: {adjustment}") print(f"Processed result: {processed_result}") print(f"Feedback result: {feedback_result}")[illegible]1. Sensor [illegible] get_data [illegible]2. MultiSensorFusion [illegible] add_sensor [illegible] fuse_data [illegible]3. AdaptiveFeedbackControl [illegible]4. ConsciousnessPerception [illegible]5. [illegible]“[illegible]”[illegible]

[illegible]● [illegible] ○ [illegible] ○ [illegible]● [illegible] ○ [illegible] ○ [illegible]● [illegible] ○ [illegible] ○ [illegible]● [illegible] ○ [illegible] ○ [illegible]● [illegible] ○ [illegible] ○ [illegible]● [illegible] ○ [illegible] ○ [illegible]

[illegible]● [illegible] ○ [illegible] X [illegible] CT [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible]

[illegible] ● [illegible] ○ [illegible] X [illegible] CT [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible] ● [illegible] ○ [illegible] Transformer [illegible] ○ [illegible] ● [illegible] ○ [illegible] GPU [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible] ● [illegible] ○ [illegible] ○ [illegible] ● [illegible] ● [illegible] ● [illegible] ● [illegible]

● [illegible] ● Midjourney [illegible] “[illegible]” [illegible] ● StableDiffusion [illegible] ● [illegible] “[illegible]” [illegible] ● IBM Watson for Oncology [illegible]

X [illegible]CT [illegible]

●[illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]● [illegible]

●[illegible]**[illegible]**[illegible]"[illegible]-[illegible]-[illegible]-[illegible]"[illegible]

```
```pythonimport numpy as npfrom typing import Dict, List, Union# === [illegible] ===class NeuroSymbolicEngine: def __init__(self): self.memory_matrix = {} # [illegible] self.concept_graph = {} # [illegible] self.meta_learning_rate = 0.01 # [illegible] def multimodal_fusion(self, inputs: Dict[str, np.ndarray]) -> np.ndarray: """[illegible] (A/B/C [illegible])""" # [illegible]/[illegible]/[illegible] (C [illegible]) purified = { 'visual': self._purify(inputs['vision'], noise_threshold=0.3), 'audio': self._extract_semantic(inputs['audio']), 'tactile': self._normalize_sensor(inputs['tactile']) } # [illegible] (A [illegible]) - [illegible] direct_feedback = self._direct_reflex(purified) # [illegible] (B [illegible]) - [illegible] indirect_feedback = self._associate_memory(purified) return np.concatenate([direct_feedback, indirect_feedback]) def _purify(self, signal: np.ndarray, noise_threshold: float) -> np.ndarray: """[illegible] (C [illegible])""" return signal * (np.abs(signal) > noise_threshold) def _extract_semantic(self, audio: np.ndarray) -> str: """[illegible] (F [illegible])""" # [illegible] (D/F [illegible]) if audio.mean() > 0.5: return "urgent" return "normal" def consciousness_generation(self, fused_input: np.ndarray) -> List[str]: """[illegible] (H [illegible])""" # [illegible] (H [illegible]) pathways = [ self._symbolic_reasoning(fused_input), # [illegible] (E [illegible]) self._neural_association(fused_input), # [illegible] self._emotional_valence(fused_input) # [illegible] ] # [illegible] (H/M [illegible]) consciousness_stream = [] for path in pathways: if path['certainty'] > 0.7: # [illegible] consciousness_stream.append(path['concept']) return consciousness_stream # [illegible] def _symbolic_reasoning(self, data: np.ndarray) -> Dict: """[illegible] (E [illegible])"""
```
```

□□□□□□ (D □) + □□□□□□ (F □) if data[0] > 0.8 and data[1] < 0.2: return {"concept": "danger_avoidance", "certainty": 0.95} return {"concept": "explore", "certainty": 0.6}# === □□□□□□ ===class AutonomousController: def __init__(self, neuro_engine: NeuroSymbolicEngine): self.neuro = neuro_engine self.feedback_loop = [] # □□□□□ def execute_cycle(self, sensor_data: dict): """□□□□□□ (□□□-□□□-□□□)""" # 1. □□□□□□□ fused = self.neuro.multimodal_fusion(sensor_data) # 2. □□□□□ consciousness = self.neuro.consciousness_generation(fused) # 3. □□□□ (□□□□□) decision = self._generalize_decision(consciousness) # 4. □□□□ (□□□) self._execute_action(decision) self._update_meta_learning(decision, sensor_data) # □□□□□ return decision def _generalize_decision(self, consciousness: list) -> str: """□□□□□□□ (□□□)""" if "danger_avoidance" in consciousness: return "emergency_stop" elif "explore" in consciousness and "curiosity" in consciousness: return "move_forward" return "standby" def _update_meta_learning(self, decision: str, sensor_data: dict): """□□□□□□ (M □□□□□)""" # □□□□□□□□□□ (M □) if decision == "emergency_stop": self.neuro.meta_learning_rate *= 1.2 # □□□□□□# === □□□□□ ===class BioSensorInterface: """□□□□□□□□□ (M □)""" def read_tactile(self) -> np.ndarray: return np.random.rand(10) # □□□□□□□ def read_audio(self) -> np.ndarray: return np.array([0.7]) # □□□□□□# === □□□□□□ ===if __name__ == "__main__": # □□□□□□□□□ neuro_engine = NeuroSymbolicEngine() controller = AutonomousController(neuro_engine) sensors = BioSensorInterface() # □□□□-□□-□□□□ for _ in range(5): sensor_data = { 'vision': np.random.rand(256,256), 'audio': sensors.read_audio(), 'tactile': sensors.read_tactile() } action = controller.execute_cycle(sensor_data) print(f"□□□□□: {action}") # □□□□: # □□□□□: emergency_stop # □□□□□: move_forward```### □□□□□□□1. **□□□□□□□□□** - □□□□□□ **A/B/C/D/E/F □□□□□□□** - □□□□□ □□□□ □□/□□□□□□ - □□□□□□□□□□□□□□□□□□□□□□□□□□□□2. **□□□□□□□** - □□`consciousness_generation()`□□ **H □□□□□** - □□□□□□□□□□□□□□ - □□□□□□□□□□□□□□B □□□□□□3. **□□□□□□□** - □□□□□□□□□□□□□□□□□□□□□□□ - □□□□□□□□□□□`meta_learning_rate`□ - □□□□□□□□□□□M □□□□□□4. **□□□□□□□** ```mermaid graph TB A[□□□□□□] --> B(□□□□□) B --> C{□□□□□□} C --> D[□□□□□□] C --> E[□□□□□□] C --> F[□□□□□□] D & E & F --> G(□□□□□) G --> H[□□□□□] H --> I[□□□□□] I --> J[□□□□] J --> C ```### □□□□□□□□1. **□□□□□□**□□□□□□□□□□Φ □□□□□□□□2. **□□□□□□**□□□□□□□□□□□□□□□□□□□3. **□□□□□□**□□□□□□□□□□□□□□□□□□4. **□□□□□□**□□□□□□□□□□□□□□□□□> □□□□□□□□□□□ **“□□□-□□□-□□□-□□□”** □□□□□□□□□□□□□□□□□□□□□/□□/□□□□□□□ `_generalize_decision()`□□□□□□□□□□□□□□□□□□□□□□□□□□□Loihi 3□□□□□□□□□□□□□□□□□

●□□□```pythonimport numpy as npimport tensorflow as tffrom tensorflow.keras.layers import Input, Dense, LSTM, Conv2D, Flatten, Concatenatefrom tensorflow.keras.models import Model# □□□□□□□□□□□□class NeuralNetworkConsciousnessSystem: def __init__(self): self.sensory_inputs = {} # □□□□ self.memory = [] # □□□□ self.current_state = None # □□□□ self.feedback = None # □□ # □□□□□□□□□ def preprocess_sensory_data(self, data, data_type): if data_type == 'visual': # □□□□□□□□□□ processed_data = self.preprocess_visual_data(data) elif data_type == 'auditory': # □□□□□□□□□□ processed_data = self.preprocess_auditory_data(data) #

□□□□□□□... return processed_data # □□□□□□ def preprocess_visual_data(self, visual_data): # □□□□□□□□□□□□□□ input_layer = Input(shape=(visual_data.shape[1:])) x = Conv2D(32, (3, 3), activation='relu')(input_layer) x = Flatten()(x) visual_model = Model(inputs=input_layer, outputs=x) return visual_model.predict(visual_data) # □□□□□□ def preprocess_auditory_data(self, auditory_data): # □□□□ LSTM □□□□□□□□ input_layer = Input(shape=(auditory_data.shape[1:])) x = LSTM(32, activation='relu')(input_layer) auditory_model = Model(inputs=input_layer, outputs=x) return auditory_model.predict(auditory_data) # □□□□□ def aggregate_sensory_data(self, processed_data_list): # □□□□□□□□□□□□□□□ input_layers = [Input(shape=(data.shape[1:])) for data in processed_data_list] concatenated = Concatenate()(input_layers) x = Dense(64, activation='relu')(concatenated) aggregation_model = Model(inputs=input_layers, outputs=x) return aggregation_model.predict(processed_data_list) # □□□□ def update_consciousness(self, aggregated_data): # □□□□□ if self.current_state is None: self.current_state = aggregated_data else: # □□ LSTM □□□□□□□ input_layer = Input(shape=(self.current_state.shape[1],)) x = Dense(64, activation='relu')(input_layer) update_model = Model(inputs=input_layer, outputs=x) self.current_state = update_model.predict([self.current_state, aggregated_data]) # □□□□ def process_feedback(self, feedback_data): # □□□□□□□ self.feedback = feedback_data # □□□□□□□□ self.update_consciousness(np.concatenate([self.current_state, self.feedback], axis=1)) # □□□□□□□ def conscious_inference(self): # □□□□□□□□ input_layer = Input(shape=(self.current_state.shape[1],)) x = Dense(32, activation='relu')(input_layer) x = Dense(16, activation='relu')(x) output = Dense(8, activation='linear')(x) inference_model = Model(inputs=input_layer, outputs=output) return inference_model.predict(self.current_state) # □□□□□□□□□ def execute_action(self, inference_result): # □□□□□□□□□□□□□□ # □□□□□□□□□□□□□□ pass# □□□□ if __name__ == "__main__": # □□□□□□□□□□□ consciousness_system = NeuralNetworkConsciousnessSystem() # □□□□□□□ visual_data = np.random.rand(1, 64, 64, 3) # □□□□□ auditory_data = np.random.rand(1, 100, 1) # □□□□□ # □□□□□□ processed_visual = consciousness_system.preprocess_visual_data(visual_data) processed_auditory = consciousness_system.preprocess_auditory_data(auditory_data) # □□□□□ aggregated_data = consciousness_system.aggregate_sensory_data([processed_visual, processed_auditory]) # □□□□□ consciousness_system.update_consciousness(aggregated_data) # □□□□□ feedback_data = np.random.rand(1, 8) consciousness_system.process_feedback(feedback_data) # □□□□ inference_result = consciousness_system.conscious_inference() # □□□□ consciousness_system.execute_action(inference_result) print("□□□□□□:", inference_result)```□□1. □□□□□□□□□□□□□□□□□□□□□□□□□□□□2. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3. □□□□□□□□□□□□□□□□□□□□□□□□□□□□4. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□5. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□6. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□7. □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□

●□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□“□□/□□□□□□□□□”□A/B □□□“□□□□□□□”□C □□□ import numpy as npimport cv2 # □□□□ import librosa # □□□□ from scipy.signal import butter, lfilter # □□□□ class MultiModalSensor: def __init__(self): self.visual_channel = True # □□□□□□□ self.audio_channel = True # □□□□□□□ self.tactile_channel = True # □□□□□□□□□□□ def read_visual(self): # □□□□□□□□□□□□□□□□□□□ return np.random.randint(0, 255, (224, 224, 3), dtype=np.uint8) def read_audio(self, fs=16000, duration=1): # □□□□□□□□□□□□□ t = np.linspace(0, duration, int(fs*duration), endpoint=False) return np.sin(2*np.pi*440*t) + 0.5*np.random.randn(len(t)) def read_tactile(self): # □□□□□□□□□□□□0-100N□ return np.random.uniform(0, 100) def filter_signal(self, signal, type='butterworth', cutoff=50): # □□□□□□□C □□□□ if type == 'butterworth': b, a = butter(4, cutoff, fs=1000, btype='lowpass') return lfilter(b, a, signal) elif type == 'median': return np.median(signal) □□□□□□□□□□□□□□□□□□□ □□“□□□□□□□”□D/E/F □□□□□□□□□□□□□D□□□□□□□□□F□□□□□□□□□□□-□□□□□□ import torchimport torch.nn as nnfrom transformers import BertTokenizer, BertModel # □□□□□□ class CognitiveFusion(nn.Module): def __init__(self, visual_dim=224*224*3, audio_dim=16000, tactile_dim=1): super().__init__() # □□□□□□CNN□ self.visual_encoder = nn.Sequential(nn.Conv2d(3, 64, 3), nn.ReLU(), nn.MaxPool2d(2), nn.Conv2d(64, 128, 3), nn.ReLU()) # □□□□□□□Transformer□ self.audio_encoder = nn.TransformerEncoder(nn.TransformerEncoderLayer(d_model=512, nhead=8), num_layers=2) # □□□□□ self.tactile_layer = nn.Linear(tactile_dim, 128) # □□□□□BERT□ self.tokenizer = BertTokenizer.from_pretrained('bert-base-uncased') self.language_model = BertModel.from_pretrained('bert-base-uncased') # □□□□□□ self.fusion_layer = nn.Linear(512+512+128, 256) def forward(self, visual_data, audio_data, tactile_data, text_input): # □□□□ visual_feat = self.visual_encoder(visual_data.unsqueeze(0)).flatten() # □□□□□□□□□+Transformer□ audio_mel = librosa.feature.melspectrogram(y=audio_data, sr=16000) audio_feat = self.audio_encoder(torch.from_numpy(audio_mel).unsqueeze(0)) # □□□□ tactile_feat = self.tactile_layer(torch.tensor([tactile_data])) # □□□□ inputs = self.tokenizer(text_input, return_tensors='pt', padding=True, truncation=True) with torch.no_grad(): language_feat = self.language_model(**inputs).last_hidden_state.mean(dim=1) # □□□□□D/E/F □□□□ fused_feat = torch.cat([visual_feat, audio_feat.flatten(), tactile_feat, language_feat], dim=-1) fused_feat = self.fusion_layer(fused_feat) return fused_feat # □□□□□□ □□□□□□□□□□□□□□□□ □□“□□□□□□□□”□①□□□“□□□□□”□H/M □□□□□□□□□□□□□□□□ import gym # □□□□□□ from stable_baselines3 import PPO # □□□□□□ class AdaptiveController: def __init__(self, state_dim, action_dim): self.env = gym.make('Pendulum-v1') # □□□□□□□□□□□ self.model = PPO("MlpPolicy", self.env, verbose=1) def train(self, timesteps=10000): self.model.learn(total_timesteps=timesteps) def predict_action(self, state): # □□□□□□□□□□□□□□□□ action, _ = self.model.predict(state, deterministic=True) return

action def feedback_loop(self, sensory_feat, action): # □□□□□□□□□□□□□□□□□□ # □□ sensory_feat □□□□□□□□□□ error = np.linalg.norm(sensory_feat - action) if error > 0.5: # □□□□□□□□□□ self.model.policy.update(error) # □□□□□□□□□□ return error □□□□□□□□□□□□□□□□□□□□□ □□□□“□□□□□□”□H □□□“□□□□□”□M □□□□□□□□□□□□□□□□ from causalnex.structure import StructureModel # □□□□ from causalnex.plots import plot_structureclass ConsciousnessSimulator: def __init__(self): self.causal_model = StructureModel() # □□□□□□□□□□□□ self.causal_model.add_edges_from([('□□', '□□□□'), ('□□', '□□□□'), ('□□', '□□□□'), ('□□□□', '□□□□'), ('□□□□', '□□□□'), ('□□□□', '□□□□')]) def infer_consciousness(self, sensory_data): # □□□□□□□□□□□□ evidence = { '□□': sensory_data['visual'], '□□': sensory_data['audio'], '□□': sensory_data['tactile'] } prediction = self.causal_model.predict(evidence) return prediction # □□□□□□□□“□□□□”“□□□□”□ def generate_commands(self, inference_result): # □□□□□□□□□□□□□□□□□□□□□□□□□□ command_map = { '□□□□': [0.5, 0.3, -0.2], # □□□□□□□□□□ '□□□□': [1.0, 0.0, 0.0], # □□□□□□□□□ '□□□□': 0.8 # □□□□□□□□□□ } return command_map.get(inference_result, [0.0, 0.0, 0.0]) □□□□□□□□□□□□ # □□□□□□ sensor = MultiModalSensor()cognitive = CognitiveFusion()controller = AdaptiveController(state_dim=3, action_dim=1)consciousness = ConsciousnessSimulator()# □□□□□□□□ while True: # 1. □□□□□ visual_data = sensor.read_visual() audio_data = sensor.read_audio() tactile_data = sensor.filter_signal(sensor.read_tactile(), cutoff=30) # □□□□□□ text_input = "□□□□□□□" # □□□□□□□□ # 2. □□□□ with torch.no_grad(): visual_tensor = torch.from_numpy(visual_data).permute(2, 0, 1).float() / 255.0 audio_tensor = torch.from_numpy(audio_data[:16000]) # □□□□□□□ fused_feat = cognitive(visual_tensor, audio_tensor, tactile_data, text_input) # 3. □□□□□□□ sensory_data = { 'visual': visual_data.mean(), 'audio': audio_data.std(), 'tactile': tactile_data } inference_result = consciousness.infer_consciousness(sensory_data) action = controller.predict_action(fused_feat.numpy()) feedback_error = controller.feedback_loop(fused_feat.numpy(), action) # 4. □□□□□□□ commands = consciousness.generate_commands(inference_result) print(f"□□□□□□: {inference_result}, □□□□: {commands}, □□□□: {feedback_error:.2f}") # □□□□□□ import time time.sleep(0.1) □□□□□□□□□□ 1. □□□“□□□”□□□ AI □□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□2. □□□□□□□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□<200ms□□□□□□□□□□□□□□□□ GPU/TPU□□- □□□□□□□□□□□□□□□□<0.01%□□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□□ AI □□□“□□□□□”□□□□□□□□□□□□3. □□□□□□□- □□□□□□□□□□□□□□□□□□□□□□□□ Intel Loihi□□□□□□□□□□□□□□□□□□□- □□□□□Embodied AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□- □□□□□□□□□“□□□□□”□□□□□□□□□□□□□□□□□□□□□□□ □□□

●□□1. □□□□□□□□□□□□□□□□□□□□□□□□□□ - □□□CNN□□ - □□2. □□□□□□□□□□□□□□□□□□□□□□□□□□ - □□□

[illegible] - [illegible] 1. [illegible] - [illegible] - [illegible] 2. [illegible] - [illegible] - [illegible] 3. [illegible] - [illegible] - [illegible]

[illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible]- [illegible] IBM [illegible]“[illegible]”[illegible]- [illegible]- [illegible]

"Artificial Intelligence Neural Network Adaptive Self-induction Self-feedback Stream of Consciousness Absorption Integration Purification Sublimation" 2025v1.1 Global Multilingual Online Edition E-book artificial intelligence technology research and development innovation peak.« Réseau neuronal d'intelligence artificielle adaptative à l'induction de l'auto-réponse du flux de conscience absorbe l'intégration, l'intégration, la purification et la sublimation » 2025v1.1 Version Web multilingue mondiale eBook R& D de la technologie de l'intelligence artificielle au sommet de l'innovation«Адаптивные нейронные сети искусственного интеллекта «Адаптивные индукции, самообменные

потоки сознания» 2025v1.1 Глобальная многоязычная сетевая версия электронной книги «Искусственный интеллект научно-исследовательская и ин новационная технология»"Red neuronal de Inteligencia Artificial Adaptive Induction Self Feedback Flujo de conciencia de absorción e integración de inte gración, Purificació n y Sublimació n "2025 v1.1 edició n Web Multilingü e Global Ebook Investigació n e Innovació n de la Tecnol Ogí a de Intelligencia Artificial ●● Neural network system involves multi-level key technologies, the core of which is to simulate the information processing mechanism of biological neural system, and at the same time, to combine engineering implementation and application requirements. The following are its core technical framework and subdivision fields: 1. Infrastructure and core algorithm 1. Neural network infrastructure-Convolutional Neural Network (CNN): used for extracting spatial features such as images and videos, and typically used in computer vision (such as image classification and object detection). -Recurrent Neural Network (RNN/LSTM/GRU): It processes sequence data (such as text and voice) and captures time-series dependencies, and is often used in natural language processing (NLP) and speech recognition. -Transformer architecture: based on self-attention mechanism, it solves the problem of long sequence dependence and becomes the core framework of NLP (such as GPT series) and multimodal models (such as BERT and CLIP). -Graph Neural Network (GNN): Processing graph structure data (such as social network and molecular structure) for recommendation system, drug research and development, etc. 2. Backpropagation, the core technology of deep learning: the basic algorithm to optimize the parameters of neural network, and update the weights through gradient descent. -Loss function and optimizer: such as cross entropy loss and mean square error (MSE). Optimizers include Adam, SGD and their variants (such as RMSprop). -Regularization technologies: Dropout, L1/L2 regularization and Batch Normalization, which are used to prevent over-fitting and improve the generalization ability of the model. Multi-modal and cross-modal fusion technology 1. Multi-modal data processing-cross-modal feature alignment: the semantic association of different modal data such as text, image and voice is realized through joint embedding space (such as image-text alignment of CLIP). -Attention mechanism extension: such as Cross-Attention and Multi-modal Transformer, which supports multi-source information interaction. -Pre-training models: such as GPT-4V (Multimodal GPT), FLAVA, and MDETR, which realize universal representation by pre-training massive multimodal data. 2. Perception layer technology-computer vision (CV): target detection (YOLO, Faster R-CNN), semantic segmentation (Mask R-CNN), 3D vision (point cloud processing, monocular vision). -Speech processing: automatic speech recognition (ASR, such as Whisper), speech synthesis (TTS, such as Tacotron), voiceprint recognition. -Natural Language Processing (NLP): word segmentation, syntactic analysis, sentiment analysis and knowledge map construction. 3. Autonomous learning and adaptive mechanism 1. Unsupervised/self-supervised learning-Contrastive Learning: Through sample similarity modeling (such as SimCLR, MoCo), the general features are learned by using unlabeled data. -Generation of countermeasure networks (GAN): used for image generation and data enhancement, with typical models such as StyleGAN and Diffusion Models. -Self-

supervised pre-training: mining the internal structure of data through mask language model (such as BERT) and automatic encoder (AE). 2. Reinforcement Learning (RL) and Adaptive Control-Deep Reinforcement Learning (DRL): Combining the DRL models of CNN/Transformer (such as DQN, PPO, SAC), it is used for robot control and autonomous driving decision. -Online learning and transfer learning: the model is continuously updated in a dynamic environment (such as incremental learning), and the old task knowledge is used to accelerate the new task learning (such as federal transfer learning). -Adaptive feedback mechanism: dynamic parameter adjustment based on environmental feedback, such as adaptive weight update and dynamic network architecture search (NAS). Model inspired by neuroscience 1. Impulsive neural network (SNN), which simulates the impulse discharge mechanism of biological neurons, has the advantages of low power consumption and time sequence processing, and is suitable for real-time sensing tasks (such as TrueNorth, a neuromorphological chip). 2. Brain-computer interface (BCI) and neural decoding-non-invasive BCI: EEG and fMRI are used to capture EEG signals and realize mind control (such as typing and wheelchair control). -Invasive BCI: Implantable electrodes directly read neuron activity (such as Neuralink) for medical rehabilitation or human-computer collaboration. 3. brain like computing architecture-Learn from the layered processing mechanism of cerebral cortex structure, such as hierarchical sequential memory (HTM) and neurocognitive machine (Neocognitron). Explanatory and transparency technology 1. Explanatory artificial intelligence (XAI)- attention visualization: show the attention area of the attention head in Transformer (such as BERT's attention analysis) through heat map. -Model interpretation tools: Shap (Shapley Additional Explanations) and Lime (Local Interpretable Model-agnostic Explanations) to analyze the decision logic of black-box model. -Symbol-Connectionism fusion: combining neural networks with rule engines (such as expert systems) to improve the traceability of decisions. 2. Causal inference-Causal diagram (such as Pearl's causal inference framework) is introduced to distinguish correlation from causality and enhance the robustness of the model. 6. Hardware and chip technology 1. Special acceleration chips-GPU/TPU/NPU: NVIDIA GPU (CUDA architecture), Google TPU (tensor processing unit), and Huawei Ascension (AI computing chip). -integrated storage and calculation chip: break the "memory wall" restriction, such as Graphcore IPU and Pinggun Technology integrated storage and calculation chip. -Brain-like chips: imitating the structure of biological neural networks, such as Intel Loihi and IBM TrueNorth. 2. Edge computing and lightweight deployment-model compression: Quantization, Pruning, Knowledge Distillation, adapting to edge devices (such as mobile phones and robots). -Real-time reasoning framework: TensorRT, ONNX Runtime, MNN, to optimize the reasoning speed of the model at the edge. Seven, data processing and feature engineering 1. Multi-source data fusion-sensor fusion technology: fusion of vision, LiDAR, IMU and other multi-sensor data (such as BEV/Transformer fusion scheme in autonomous driving). -Spatio-temporal data processing: Time series feature extraction for video and trajectory data (such as I3D and LSTM-CNN mixed model). 2. Self-monitoring data enhancement-automatically generate enhancement strategies (such as auto-augmentation), and improve the robustness of the model through confrontation samples (FGSM).

8. Combination of reinforcement learning and control theory 1. Adaptive control algorithm-Model Predictive Control (MPC) is combined with DRL for robot path planning and industrial automation control. -Adaptive PID control: dynamically adjust PID parameters through neural network to optimize the response speed of the system. 2. Physical world interaction-Model-based reinforcement learning (RL): The dynamic model is used to predict the environmental state and reduce the trial and error cost of the real environment (such as the simulation environment in robot training). 9. Ethics and security technology 1. Adversarial Training and sample detection to improve the robustness of the model to malicious input. -Robustness evaluation criteria: such as attack detection indicators in CleverHans library. 2. Privacy protection technology-Federated Learning: collaborative training model (such as medical data scene) with data not local. -Differential Privacy: adding noise in data collection and modeling to protect individual privacy. X. Frontier Exploration: General Intelligence and Self-consciousness 1. Theoretical Attempt of Consciousness Modeling-Neural Network Architecture Based on Global Workspace Theory (GWT) to simulate the brain's attention and information integration mechanism. -Dynamic system theory: neural network is regarded as a nonlinear dynamic system, and its emergent behavior is studied (such as the application of chaos theory in neurodynamics). 2. The engineering path of autonomous consciousness-Meta-Learning: let the model learn "how to learn" and simulate the ability of human beings to adapt to new tasks quickly (such as MAML algorithm). -Intrinsic motivation-driven learning: exploring the environment independently through Curiosity-Driven Learning model, rather than relying on external rewards. Summary: Technical cooperation and challenge The breakthrough of artificial intelligence neural network depends on the synergy of algorithm innovation (such as more efficient attention mechanism), hardware upgrade (such as brain-like chip), data revolution (such as high-quality multimodal data set) and theoretical breakthrough (such as mathematical modeling of consciousness). The current challenges include:-the generalization bottleneck of general intelligence: how to make the model realize human-like reasoning and decision-making in an open environment; -Energy consumption and scalability: training cost and environmental impact of large-scale models (such as trillion-parameter GPT); -Ethics and safety: social problems such as the attribution of decision-making responsibility and algorithm bias in autonomous systems. In the future, the cross study of neuroscience, computer science and philosophy may become the key to solve the puzzle of "autonomous consciousness", while the engineering landing needs to focus on the adaptation of scene technology in vertical fields (such as medical care and industry). ● Research and development of artificial intelligence technology: large model, multi-model and multi-modal general agent, high intelligent robot, high intelligent automatic driving, high intelligent industrial manufacturing, high intelligent telemedicine, high intelligent education and learning, high intelligent financial management, high intelligent agricultural engineering, high intelligent medical care and old-age robot, and so on. As a result, it is very important to integrate and generalize all kinds of advanced technologies. However, the foundation of the research and development of artificial intelligence technology lies in the in-depth and meticulous industrial

wisdom revolution, especially the generalization and expansion of the autonomous consciousness of neural network system, which is very important, because it is the core code technology of artificial intelligence, which goes beyond the scope of the previous artificial intelligence neural network system. From machinery to machinery, the perception of singularity is controlled by inductive feedback, rather than the self-adaptive self-induction, self-feedback and self-generalization thinking mode of artificial intelligence neural network system. ① Self-adaptive, self-feedback, self-response and self-control. Although the brain-computer interface is very important, there is no doubt that it needs perfect and accurate feedback of biological control between man and machine, otherwise there will be various drawbacks. ② The stream of consciousness in artificial intelligence neural network system is mainly sensory, tactile and auditory vision, including a full set of information, audio, video, text, pictures and images, etc. Accepting and adapting to perceptual cognitive feedback consciousness A. Directly absorbing feedback consciousness B. Indirectly absorbing feedback consciousness C. Filtering, purifying and deepening mechanical signals D. Mathematical coding program E. Image language logical thinking F. Natural language processing D. Hybrid logic language, mixed language image language Mathematical logic language natural language processing, identification, purification and deepening, which is the success or failure of conscious cognitive absorption feedback in artificial intelligence advanced neural network system. Otherwise, the mechatronics technology and the advanced nervous system consciousness adaptive self-induction, self-cognition, self-perception, self-filtering, self-reaction and self-feedback self-control system of human brain will inevitably lose their basic relevance and contact. H. Simple physical and chemical information signals will not directly produce the shallowest hazy stream of consciousness, and it is difficult to realize basic feedback consciousness no matter how they interact. The key lies in multi-channel opening, comprehensive integration and optimization of various channels.

●"Artificial Intelligence Neural Network Adaptive Self-induction Self-feedback Stream of Consciousness Absorption Integration Purification Sublimation" 2025v1.1 Global Multilingual Online Edition E-book artificial intelligence technology research and development innovation peak.« Réseau neuronal d'inte lligence artificielle adaptative à l'induction de l'auto-réponse du flux de conscience absorbe l'intégration, l'intégration, la purification et la sublimation » 2025v1.1 Version Web multilingue mondiale eBook R& D de la technologie de l'intelligence artificielle au sommet de l'innovation«Адаптивные нейронные сети искусственного интеллекта «Адаптивные индукции, самообменные потоки сознания» 2025v1.1 Глобальная многоязычная сетевая версия электронной книги «Искусственный интеллект научно-исследовательская и ин новационная технология»"Red neuronal de Inteligencia Artificial Adaptive Induction Self Feedback Flujo de conciencia de absorción e integración de inte gración, Purificació n y Sublimació n "2025 v1.1 edició n Web Multilingü e Global Ebook Investigació n e Innovació n de la Tecnol Ogí a de Intelligencia Artificial
●● Neural network system involves multi-level key technologies, the core of

which is to simulate the information processing mechanism of biological neural system, and at the same time, to combine engineering implementation and application requirements. The following are its core technical framework and subdivision fields: 1. Infrastructure and core algorithm 1. Neural network infrastructure-Convolutional Neural Network (CNN): used for extracting spatial features such as images and videos, and typically used in computer vision (such as image classification and object detection). -Recurrent Neural Network (RNN/LSTM/GRU): It processes sequence data (such as text and voice) and captures time-series dependencies, and is often used in natural language processing (NLP) and speech recognition. -Transformer architecture: based on self-attention mechanism, it solves the problem of long sequence dependence and becomes the core framework of NLP (such as GPT series) and multimodal models (such as BERT and CLIP). -Graph Neural Network (GNN): Processing graph structure data (such as social network and molecular structure) for recommendation system, drug research and development, etc. 2. Backpropagation, the core technology of deep learning: the basic algorithm to optimize the parameters of neural network, and update the weights through gradient descent. -Loss function and optimizer: such as cross entropy loss and mean square error (MSE). Optimizers include Adam, SGD and their variants (such as RMSprop). -Regularization technologies: Dropout, L1/L2 regularization and Batch Normalization, which are used to prevent over-fitting and improve the generalization ability of the model. Multi-modal and cross-modal fusion technology 1. Multi-modal data processing-cross-modal feature alignment: the semantic association of different modal data such as text, image and voice is realized through joint embedding space (such as image-text alignment of CLIP). -Attention mechanism extension: such as Cross-Attention and Multi-modal Transformer, which supports multi-source information interaction. -Pre-training models: such as GPT-4V (Multimodal GPT), FLAVA, and MDETR, which realize universal representation by pre-training massive multimodal data. 2. Perception layer technology-computer vision (CV): target detection (YOLO, Faster R-CNN), semantic segmentation (Mask R-CNN), 3D vision (point cloud processing, monocular vision). -Speech processing: automatic speech recognition (ASR, such as Whisper), speech synthesis (TTS, such as Tacotron), voiceprint recognition. -Natural Language Processing (NLP): word segmentation, syntactic analysis, sentiment analysis and knowledge map construction. 3. Autonomous learning and adaptive mechanism 1. Unsupervised/self-supervised learning-Contrastive Learning: Through sample similarity modeling (such as SimCLR, MoCo), the general features are learned by using unlabeled data. -Generation of countermeasure networks (GAN): used for image generation and data enhancement, with typical models such as StyleGAN and Diffusion Models. -Self-supervised pre-training: mining the internal structure of data through mask language model (such as BERT) and automatic encoder (AE). 2. Reinforcement Learning (RL) and Adaptive Control-Deep Reinforcement Learning (DRL): Combining the DRL models of CNN/Transformer (such as DQN, PPO, SAC), it is used for robot control and autonomous driving decision. -Online learning and transfer learning: the model is continuously updated in a dynamic environment (such as incremental learning), and the old task knowledge is used to accelerate

the new task learning (such as federal transfer learning). -Adaptive feedback mechanism: dynamic parameter adjustment based on environmental feedback, such as adaptive weight update and dynamic network architecture search (NAS). Model inspired by neuroscience 1. Impulsive neural network (SNN), which simulates the impulse discharge mechanism of biological neurons, has the advantages of low power consumption and time sequence processing, and is suitable for real-time sensing tasks (such as TrueNorth, a neuromorphological chip). 2. Brain-computer interface (BCI) and neural decoding-non-invasive BCI: EEG and fMRI are used to capture EEG signals and realize mind control (such as typing and wheelchair control). -Invasive BCI: Implantable electrodes directly read neuron activity (such as Neuralink) for medical rehabilitation or human-computer collaboration. 3. brain like computing architecture-Learn from the layered processing mechanism of cerebral cortex structure, such as hierarchical sequential memory (HTM) and neurocognitive machine (Neocognitron). Explanatory and transparency technology 1. Explanatory artificial intelligence (XAI)- attention visualization: show the attention area of the attention head in Transformer (such as BERT's attention analysis) through heat map. -Model interpretation tools: Shap (Shapley Additional Explanations) and Lime (Local Interpretable Model-agnostic Explanations) to analyze the decision logic of black-box model. -Symbol-Connectionism fusion: combining neural networks with rule engines (such as expert systems) to improve the traceability of decisions. 2. Causal inference-Causal diagram (such as Pearl's causal inference framework) is introduced to distinguish correlation from causality and enhance the robustness of the model. 6. Hardware and chip technology 1. Special acceleration chips-GPU/TPU/NPU: NVIDIA GPU (CUDA architecture), Google TPU (tensor processing unit), and Huawei Ascension (AI computing chip). -integrated storage and calculation chip: break the "memory wall" restriction, such as Graphcore IPU and Pinggun Technology integrated storage and calculation chip. -Brain-like chips: imitating the structure of biological neural networks, such as Intel Loihi and IBM TrueNorth. 2. Edge computing and lightweight deployment-model compression: Quantization, Pruning, Knowledge Distillation, adapting to edge devices (such as mobile phones and robots). -Real-time reasoning framework: TensorRT, ONNX Runtime, MNN, to optimize the reasoning speed of the model at the edge. Seven, data processing and feature engineering 1. Multi-source data fusion-sensor fusion technology: fusion of vision, LiDAR, IMU and other multi-sensor data (such as BEV/Transformer fusion scheme in autonomous driving). -Spatio-temporal data processing: Time series feature extraction for video and trajectory data (such as I3D and LSTM-CNN mixed model). 2. Self-monitoring data enhancement-automatically generate enhancement strategies (such as auto-augmentation), and improve the robustness of the model through confrontation samples (FGSM). 8. Combination of reinforcement learning and control theory 1. Adaptive control algorithm-Model Predictive Control (MPC) is combined with DRL for robot path planning and industrial automation control. -Adaptive PID control: dynamically adjust PID parameters through neural network to optimize the response speed of the system. 2. Physical world interaction-Model-based reinforcement learning (RL): The dynamic model is used to predict the environmental state and reduce the trial and error cost of the real environment (such as the simulation

environment in robot training). 9. Ethics and security technology 1. Adversarial Training and sample detection to improve the robustness of the model to malicious input. -Robustness evaluation criteria: such as attack detection indicators in CleverHans library. 2. Privacy protection technology-Federated Learning: collaborative training model (such as medical data scene) with data not local. -Differential Privacy: adding noise in data collection and modeling to protect individual privacy. X. Frontier Exploration: General Intelligence and Self-consciousness 1. Theoretical Attempt of Consciousness Modeling-Neural Network Architecture Based on Global Workspace Theory (GWT) to simulate the brain's attention and information integration mechanism. -Dynamic system theory: neural network is regarded as a nonlinear dynamic system, and its emergent behavior is studied (such as the application of chaos theory in neurodynamics). 2. The engineering path of autonomous consciousness-Meta-Learning: let the model learn "how to learn" and simulate the ability of human beings to adapt to new tasks quickly (such as MAML algorithm). -Intrinsic motivation-driven learning: exploring the environment independently through Curiosity-Driven Learning model, rather than relying on external rewards. Summary: Technical cooperation and challenge The breakthrough of artificial intelligence neural network depends on the synergy of algorithm innovation (such as more efficient attention mechanism), hardware upgrade (such as brain-like chip), data revolution (such as high-quality multimodal data set) and theoretical breakthrough (such as mathematical modeling of consciousness). The current challenges include:-the generalization bottleneck of general intelligence: how to make the model realize human-like reasoning and decision-making in an open environment; -Energy consumption and scalability: training cost and environmental impact of large-scale models (such as trillion-parameter GPT); -Ethics and safety: social problems such as the attribution of decision-making responsibility and algorithm bias in autonomous systems. In the future, the cross study of neuroscience, computer science and philosophy may become the key to solve the puzzle of "autonomous consciousness", while the engineering landing needs to focus on the adaptation of scene technology in vertical fields (such as medical care and industry). ● Research and development of artificial intelligence technology: large model, multi-model and multi-modal general agent, high intelligent robot, high intelligent automatic driving, high intelligent industrial manufacturing, high intelligent telemedicine, high intelligent education and learning, high intelligent financial management, high intelligent agricultural engineering, high intelligent medical care and old-age robot, and so on. As a result, it is very important to integrate and generalize all kinds of advanced technologies. However, the foundation of the research and development of artificial intelligence technology lies in the in-depth and meticulous industrial wisdom revolution, especially the generalization and expansion of the autonomous consciousness of neural network system, which is very important, because it is the core code technology of artificial intelligence, which goes beyond the scope of the previous artificial intelligence neural network system. From machinery to machinery, the perception of singularity is controlled by inductive feedback, rather than the self-adaptive self-induction, self-feedback and self-generalization thinking mode of artificial intelligence neural network

system. ① Self-adaptive, self-feedback, self-response and self-control. Although the brain-computer interface is very important, there is no doubt that it needs perfect and accurate feedback of biological control between man and machine, otherwise there will be various drawbacks. ② The stream of consciousness in artificial intelligence neural network system is mainly sensory, tactile and auditory vision, including a full set of information, audio, video, text, pictures and images, etc. Accepting and adapting to perceptual cognitive feedback consciousness A. Directly absorbing feedback consciousness B. Indirectly absorbing feedback consciousness C. Filtering, purifying and deepening mechanical signals D. Mathematical coding program E. Image language logical thinking F. Natural language processing D. Hybrid logic language, mixed language image language Mathematical logic language natural language processing, identification, purification and deepening, which is the success or failure of conscious cognitive absorption feedback in artificial intelligence advanced neural network system. Otherwise, the mechatronics technology and the advanced nervous system consciousness adaptive self-induction, self-cognition, self-perception, self-filtering, self-reaction and self-feedback self-control system of human brain will inevitably lose their basic relevance and contact. H. Simple physical and chemical information signals will not directly produce the shallowest hazy stream of consciousness, and it is difficult to realize basic feedback consciousness no matter how they interact. The key lies in multi-channel opening, comprehensive integration and optimization of various channels.